

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
Факультет електроніки
(повна назва інституту/факультету)

Кафедра акустичних та мультимедійних електронних систем
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

 С.А. Найда
(підпис) (ініціали, прізвище)

“01” червня 2020 р.

Дипломна робота
на здобуття ступеня бакалавра

зі спеціальності 171 Електроніка (Електронні та інформаційні системи і
технології телебачення, кінематографії та звукотехніки).

на тему: «Системи голосового керування функціональним обладнанням
“розумного” будинку»

Виконав: студент IV курсу, групи ДВ-61
(шифр групи)

Петровський Владислав Олександрович
(прізвище, ім'я, по батькові)


(підпис)

Керівник

д.т.н., професор Власюк Г.Г
(посада, науковий ступінь, вчене звання, прізвище та ініціали)


(підпис)

Консультант

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент доцент кафедри ЕПС, к.т.н., Михайлов С.Р
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ініціали)


(підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авто-
рів без відповідних посилань.

Студент
(підпис)



Київ – 2020 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет Електроніки

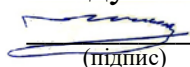
Кафедра акустичних та мультимедійних електронних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 171 Електроніка (Електронні та інформаційні системи і технології телебачення, кінематографії та звукотехніки)

ЗАТВЕРДЖУЮ

Завідувач кафедри


(підпис)

С.А. Найда
(ініціали, прізвище)

«25» травня 2020 р.

**ЗАВДАННЯ
на дипломну роботу студенту**

Петровському Владиславу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Системи голосового керування функціональним обладнанням “розумного” будинку»

керівник роботи Власюк Ганна Григорівна, д.т.н., професор.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «25»травня 2020 р. №1196-с

2. Строк подання студентом роботи 01 червня 2020 р.

3. Вихідні дані до роботи: 1) Протокол MQTT; 2)Протокол XMPP; 3) протокол AMPQ 4) Google Cloud Speech API 5) Amazon Alexa 6) Модель SnowBoy.

4. Зміст дипломної роботи: 1) Розглянути основні вимоги та стандарти для системи розумний будинок; 2) Розглянути архітектури та основні технічні особливості кожної технології; 3) Дослідити протоколи передавання даних для вибору протоколу обміну; 4) Розробити ПЗ для “розумного” будинку; 5) Оцінити переваги та недоліки кожної технології.



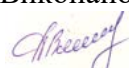

5. Перелік ілюстративного матеріалу: комплект презентації за матеріалами проведеного дослідження.

6. Консультанти розділів проекту (роботи)*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 11 березня 2020 р

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Написання першого розділу	24.03.2020	Виконано 
2	Написання другого розділу	20.04.2020	Виконано 
3	Написання третього розділу	10.05.2020	Виконано 
4	Підготовка матеріалів до друку та оформлення пояснювальної записки	30.05.2020	Виконано 
5	Підготовка та оформлення презентації для доповіді	1.06.2020	Виконано 

Студент



(підпис)

В.О Петровський

(ініціали, прізвище)

Керівник роботи



(підпис)

Г.Г Власюк

(ініціали, прізвище)

УДК 621.39621.5

РЕФЕРАТ

Дипломна робота: 61с., 19 рис., 1 дод., 28 джерел.

ГОЛОСОВЕ КЕРУВАННЯ, МОБІЛЬНИЙ ДОДАТОК, РОЗУМНИЙ БУДИНОК.

Об’єктом дослідження є технології та стандарти систем голосового керування функціональним обладнанням “розумного” будинку.

Метою дипломної роботи є дослідження систем голосового керування, які покращують життя людей з обмеженими можливостями здоров’я, та розроблення мобільного додатка для платформи Android з функцією голосового управління системою «Розумний будинок».

Актуальність теми обумовлено бурхливим розвитком голосових систем та мереж, що забезпечують швидке виконання будь-якої роботи.

Результатом дипломної роботи систематизація та узагальнення світового досвіду стосовно розвитку систем голосового керування в цілому, формування порівняльних характеристик різних систем голосового керування спеціалізованими та альтернативними системами, з’ясування позитивних та негативних властивостей різних способів передачі голосу, та готовий додаток для платформи Android, за допомогою якого можливе керування домашніми приладами, такими як чайник, мультимедія, телевізор.

Abstract

The object of research is the technology and standards of voice control systems for functional equipment of a "smart" house.

The purpose of the thesis is to study voice control systems that improve lives for people with disabilities, and to develop a mobile application for the Android platform with voice control of the "Smart Home" system. The result of the thesis is the systematization of the functionality of television systems that provide content maintenance, the formation of comparative characteristics in relation to different ways of transmitting content to television systems, to find out the positive and negative properties of different ways of content transfer.

The result of the thesis systematizes and generalizes the world experience in the development of voice control systems in general, the formation of comparative characteristics of different voice control systems with specialized and alternative systems, clarifying the positive and negative properties of different methods of voice transmission, and a ready application for Android platform. it is possible to control household appliances such as kettle, multimedia, TV.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	8
ВСТУП.....	9
1 СИСТЕМИ ГОЛОСОВОГО КЕРУВАННЯ ФУНКЦІОНАЛЬНИМ ОБЛАДНАННЯМ “РОЗУМНОГО” БУДИНКУ.....	10
1.1 Огляд предметної області.....	10
1.2 Критерії порівняння систем голосового управління	11
1.3 Огляд аналогів розроблюваної системи	12
1.3.1 Сімейство Amazon Echo.....	12
1.3.2 Google Home	13
1.3.3 Nest Learning Thermostat	15
1.3.4 Apple Siri.....	16
1.4 Висновки до розділу 1	18
2 ПРИНЦИПИ, ТЕХНІЧНІ ПАРАМЕТРИ ТА ХАРАКТЕРИСТИКИ НАЯВНИХ СИСТЕМ ГОЛОСОВОГО КЕРУВАННЯ.....	19
2.1 Функції і підсистеми додатку	19
2.2 Підсистема голосового інтерфейсу	20
2.2.1 Аналіз технологій розпізнавання мовленн	21
2.2.2 Критерії оцінки	22
2.2.3 Мовні технології SpeechKit.....	22
2.2.4 Google Cloud Speech API.....	23
2.2.5 Alexa Voice Service	24
2.2.6 SnowBoy	24
2.2.7 Тестування систем розпізнавання мови	25
2.3 Наявність апаратних модулів, їх структури та особливостей програмування, голосового керування функціональним обладнанням “розумного” будинку.....	27
2.3.1 MQTT.....	27
2.3.2 XMPP.....	29
2.3.3 AMQP	30
2.4 Висновки до розділу 2.....	31
3 ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ, ВПРОВАДЖЕННЯ, ТА БЕЗПЕКИ СИСТЕМ ГОЛОСОВОГО КЕРУВАННЯ ФУНКЦІОНАЛЬНИМ ОБЛАДНАННЯМ “РОЗУМНОГО” БУДИНКУ	32

3.1 Реалізація голосового інтерфейсу.....	33
3.1.1 Алгоритм роботи голосового інтерфейсу.....	33
3.2 Реалізація інтерфейсу до апаратної частини.....	34
3.2.1 Взаємодія з протоколом.....	34
3.3 Реалізація бази даних	34
3.3.1 Описання роботи бази даних.....	35
3.4 Реалізація підсистеми виконання команд	36
3.5 Висновки до розділу 3.....	37
4 ПРИКЛАДИ ВИКОРИСТАННЯ СИСТЕМ ГОЛОСОВОГО КЕРУВАННЯ ФУНКЦІОНАЛЬНИМ ОБЛАДНАННЯМ РОЗУМНОГО БУДИНКУ, ТА ІНТЕРФЕЙС СИСТЕМИ	38
4.1 Інтерфейс адміністратора.....	38
4.1.1 Налаштування додатку	39
4.1.2 Налаштування MQTT	40
4.1.3 Налаштування додатку	41
4.1.4 Налаштування топіків.....	41
4.1.5 Налаштування значень за замовчування	45
4.2 Інтерфейс клієнта	46
4.2.1 Обране	47
4.2.2 Інтерфейс підключення	48
4.3 Тестування додатку	49
4.4 Висновки до розділу 4.....	50
ВИСНОВКИ	51
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	53
ДОДАТОК А. КЛАС ОБРОБКИ КОМАНД.....	56
ДОДАТОК Б. SUMMARY	59

ПЕРЕЛІК СКОРОЧЕНЬ

API	– Application Programming Interface;
LAN	– Local Area Network;
HTTP	– Advanced Message Queuing Protocol;
AMQP	– Simple Mail Transfer Protocol;
AVS	– Alexa Voice Service;
JS	– JavaScript;
MQTT	– Message Queuing Telemetry Transport;
XMPP	– eXtensible Messaging and Presence Protocol;
XML	– eXtensible Markup Language;
GigE	– Gigabit Ethernet;
EPG	– Electronic Program Guide;
ПК	– Персональний комп'ютер;
OS	– Operating System
APP	– Application;
SSL	– Secure Sockets Layer;
IoT	– Internet of Things;
Wi-Fi	– Wireless Fidelity;
TCP	– Transmission Control Protocol;
HDMI	– High-Definition Multimedia Interface;

ВСТУП

Системи «розумний будинок» швидко розвиваються, вони використовуються для забезпечення комфорту, зручності, якості життя та безпеки для мешканців. В даний час більшість систем, використовуються для зменшення людської праці. Система «розумний будинок» може бути спроектована та розроблена за допомоги одного контролера, який має можливість керувати та контролювати різні пов'язані між собою пристрої, такі як штепсельні вилки, світильники, датчики температури та вологості, димові, газові та пожежні сповіщувачі, а також аварійні та охоронні системи. Одна з найбільших переваг системи полягає в тому, що можна легко керувати та управляти масивом пристроїв, таких як смартфон, планшет, настільний комп'ютер та ноутбук. На даний час розробка систем «розумний будинок» є дуже актуальними по всьому світу і я впевнений, що через 10 років буде більше домівок з системою «розумний дім», аніж звичайних. Удосконалення швидкодії розпізнавання голосових команд існуючої бібліотеки, за рахунок використання словників та асинхронних методів програмування та впровадження її в комп'ютеризовану систему «Розумний будинок», яка повинна бути розроблена за сучасною трьох рівневою архітектурою. Як запрограмувати ці технології в «розумний будинок», розробити додаток для мобільного, і присвячена дана дипломна робота.

1 СИСТЕМИ ГОЛОСОВОГО КЕРУВАННЯ ФУНКЦІОНАЛЬНИМ ОБЛАДНАННЯМ “РОЗУМНОГО” БУДИНКУ

1.1 Огляд предметної області

Розглядаючи систему «Розумний будинок», можна виділити кілька типів інтерфейсів для взаємодії з користувачем:

- кнопочний або сенсорний інтерфейс у вигляді пульта управління;
- сенсорна панель з вбудованим програмним забезпеченням;
- додаток, яке встановлюється на призначений для користувача персональний комп’ютер або смартфон;

Сенсорна панель є не дуже зручним інтерфейсом, так як з’являється необхідність встановлювати його в кожній кімнаті для зручності користування, інакше необхідно переходити в ту кімнату, де він встановлений.

Пульт управління також не дуже хороший у взаємодії, тому що позбавлений функцій віддаленого управління «Розумним будинком», наприклад, можливості дистанційно його вимкнути або зняти охорону.

Розглядаючи додаток також можна виділити 2 типу реалізації:

- екранний інтерфейс, де користувач повинен натиснути на певну піктограму або ввести текст для виконання дії;
- голосовий інтерфейс, який дозволяє розпізнавати голосовий текст і на основі результатів виконувати необхідні дії.

Голосовий інтерфейс - це програмний продукт, який за допомогою голосової або мовної платформи дозволяє взаємодіяти користувачеві з системою «Розумний будинок», віддаючи команди для запуску окремих сценаріїв і дій голосом і отримуючи інформацію в формі звукових повідомлень. Відповідно, основними специфічними функціями таких інтерфейсів є розпізнавання голосових команд і синтез звукових повідомлень .

З розвитком технологій голосовий інтерфейс став все більш поширеним, людина все частіше користується перевагами цієї безконтактної технології .

Голосові інтерфейси зручні, коли вводити текст складно або незручно. Наприклад, під час водіння автомобіля користувач може проговорити свій запит, продиктувати потрібну адресу, перевірити пробки в додатку навігатора. Або ж якщо користувач виконує дуже багато завдань і не може сконцентруватися на одній.

Можливим рішенням реалізації голосового інтерфейсу всередині приміщення могло б бути розміщення мікрофонів і джерел звуку в різних точках приміщення, таким чином, щоб користувач міг у будь-який момент вимовити команду і почути відповідь. Однак спроби реалізувати такі системи зіткнулися з тим, що було потрібно велику кількість чутливих мікрофонів, тому система ставала технічно складною і дорогою. У той же час, сучасна людина практично постійно носить з собою мобільний телефон, це робить реалізації голосового інтерфейсу набагато дешевше, а також додає можливість віддаленого контролю. Тому реалізація мобільного додатка є більш зручним рішенням для голосового управління «Розумним будинком».

Проект по розробці мобільного додатку спрямований саме на голосове управління «Розумним будинком»,

1.2 Критерії порівняння систем голосового управління

Системи голосового управління, використовувані в розумному будинку, повинні відповідати таким критеріям:

- створення власних команд;
- підтримка російської мови;
- можливість зворотного взаємодії з користувачем;

1.3 Огляд аналогів розроблюваної системи

1.3.1 Сімейство Amazon Echo

Amazon Echo - серія «розумних» пристроїв, розроблених компанією Amazon. Перше покоління пристроїв, з даної лінійки, з'явилося в 2014 році з виходом Bluetooth-колонки Amazon Echo. Amazon Echo - пристрої, створені як голосові помічники, які працюють в режимі "AlwaysOn". Взаємодія з ними відбувається тільки голосом, без використання кнопок. В колонках використовується власна технологія розпізнавання мови Alexa. Дана лінійка пристроїв може керувати пристроями розумного будинку, відтворювати музику з телефону, Amazon Prime або Spotify, а також замовляти продукти з Amazon.

В Echo другого покоління заявлено краще звуковідтворення завдяки системі Dolby, яка була представлена влітку 2016 року - це дозволяє використовувати мультікомнатну аудіосистему.



Рисунок 1.2 – Голосовий помічник від Amazon «Alexa»

У гаджета Amazon Echo Show доданий тачскрін, зображений на рис. 1.2, камера і розумний спікер, що дозволяє робити відеодзвінки, переглядати відеокамери, дивитися Amazon Prime Video. На 2018 рік компанія випустила понад 10 пристроїв для розумного будинку. Ціна стартує з 36 \$.

Переваги:

- можливість голосового оповіщення з використанням колонок;

Недоліки:

- не має підтримки російської мови;
- для створення власних команд необхідне знання програмування;
- висока вартість;

1.3.2 Система Google Home

Google Home - бездротовий динамік, зображений на рис. 1.3, забезпечений голосовим управлінням, розроблений в Google. Продукт був оголошений 18 травня 2016 року у Google I / O. Являє собою перший смарт-динамік Google. Також став одним з пристроїв, що підтримують роботу персонального асистента «Google Assistant», поряд з текстовим чатом Allo і відеочатом Duo.

Продукт дуже схожий за своєю концепцією на Amazon Echo, і є його прямим конкурентом в промисловості «розумних» колонок і персональних помічників. Також частина функціональності схожа на голосовий помічник Apple Siri [6].

Вартість від 99 \$.

Переваги:

- легкість інтеграції в існуючі рішення;

Недоліки:

- висока вартість



Рисунок 1.3 – Голосовий помічник Google Home

1.3.3 Система Nest Learning Thermostat

Nest Learning Thermostat – це розумний термостат, розроблений Nest Labs (рисунок 1.4). Це електронний, програмований пристрій, який здатен до навчання. Пристрій із підтримкою Wi-Fi, який оптимізує опалення та охолодження будинків та підприємств для економії енергії.



Рисунок 1.4 – Вигляд системи Nest Learning Thermostat

Заснована на алгоритмі машинного навчання: протягом перших тижнів користувачі повинні регулювати термостат, щоб забезпечити набір еталонних даних. Після чого Nest Learning Thermostat навчитися графіку людей, яку температурі вони використовують і коли. Використовуючи вбудовані, він може перейти в режим енергозбереження, коли він усвідомлює, що ніхто не вдома.

Переваги:

- Використання алгоритму машинного навчання;
- Мала ціна;
- Ручне керування через Wi-fi;

Недоліки:

- Не масштабується;
- Не підтримує сторонні засоби управління;
- Малий функціонал;
- Відсутність вбудованої системи розпізнавання голосових команд;

1.3.4 Програмний засіб Apple Siri

Apple Siri - хмарний персональний помічник і питально-відповідна система, програмний клієнт якої входить до складу iOS, watchOS, macOS, і tvOS компанії Apple. Цей додаток використовує обробку природної мови, щоб відповідати на питання і давати рекомендації. Siri пристосовується до кожного користувача індивідуально, вивчаючи його переваги протягом довгого часу.

Спочатку Siri стала доступною в App Store як додаток для iOS від Siri Inc. Незабаром, 28 квітня 2010 року, Siri Inc. була придбана Apple Inc. Але ще до того, як Apple купила Siri, було оголошено, що їх програмне забезпечення буде доступно для телефонів BlackBerry і телефонів під управлінням Android, потім ці плани були скасовані через покупки.

Для взаємодії Siri і «Розумного будинку» компанія Apple створила проект HomeKit.

Дана технологія має такі можливості:

- siri може регулювати роботу аксесуарів HomeKit, наприклад світлових приладів або термостата;
- siri може регулювати роботу аксесуарів HomeKit, наприклад світлових

приладів або термостата;

- Siri знає, які аксесуари HomeKit налаштовані в програмі «Дім», а також їх статус.
- Siri ідентифікує аксесуари за їхніми іменами, розташування і, за іншими відомостями, зазначеними для них в програмі «Дім». Якщо налаштований HomePod (зображений на рис. 1.5), Apple TV або iPad в якості домашнього центру, то можна використовувати Siri для управління будинком в будь-якому місці;
- якщо аксесуари організовані по кімнатах або зонам, можна управляти областями будинку однією командою;
- при використанні HomePod є можливість управляти всіма аксесуарами HomeKit в кімнаті, де знаходиться цей пристрій, однією командою
- сценарії дозволяють управляти декількома аксесуарами одночасно. За допомогою Siri можна налаштувати сценарій, використовуючи тільки голосові команди
- перевірка стану будинку
- починаючи з версії iOS 8.3 є підтримка російської мови
- переваги даної технології
- безкоштовний для використання.

Недоліки:

- необхідно використовувати тільки пристрої від компанії Apple.
- для роботи можливо використовувати тільки сертифіковані компанією Apple компоненти для роботи «Розумного будинку».



Рисунок 1.5 – Вигляд системи Apple Homepod

1.4 Висновки до розділу 1

Розглянуті реалізації є закритими і дорогими для інтеграції в «Розумний будинок». Доцільним є рішення про створення власної доступної і дешевої реалізації голосового інтерфейсу, який взаємодіє з уже існуючою реалізацією «Розумного будинку». За допомогою такого інтерфейсу можливо контролювати всі пристрої, які приєднані до однієї мережі Wi-fi, безпосередньо зі свого телефону.

2 ПРИНЦИПИ, ТЕХНІЧНІ ПАРАМЕТРИ ТА ХАРАКТЕРИСТИКИ НАЯВНИХ СИСТЕМ ГОЛОСОВОГО КЕРУВАННЯ

2.1 Функції і підсистеми додатку

В даному мобільному додатку будуть реалізовані такі функції:

- функція автоматичного старту програми при старті операційної системи;
- функція розпізнавання голосових команд;
- функція зворотного зв'язку з користувачем у вигляді синтезу голосу або виведення повідомлень на екран;
- функція з'єднання і управління «Розумним будинком»;
- функція настройки інтерфейсу додатку;
- функція адміністрування програми;

Функція розпізнавання голосових команд є основною функцією мобільного додатка. Функція налаштування інтерфейсу додатку містить функціонал графічної налаштування програми із застосуванням пунктів меню. Функція адміністрування програми дозволяє здійснювати первинне налаштування програми та подальшої зміни параметрів.

У мобільному додатку були виділені такі підсистеми:

- голосовий інтерфейс;
- інтерфейс до апаратної частини, що включає в себе функцію з'єднання з «Розумним будинком»;
- збереження даних;
- підсистема виконання команд;
- користувальницький інтерфейс.

В результаті аналізу функцій запропоновано структурну схему, зображена наведену на рис. 1.6.

Мобільний додаток

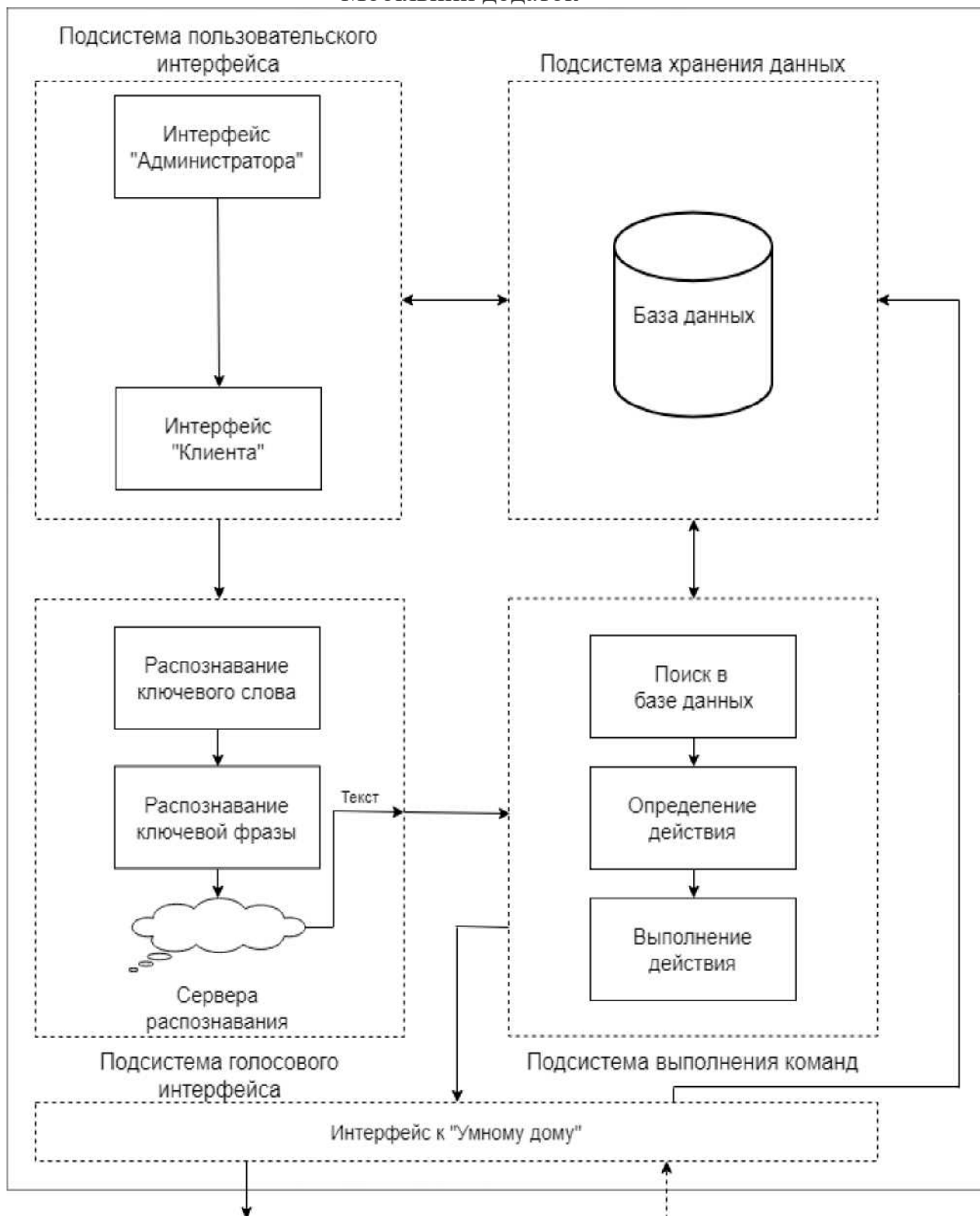


Рисунок 1.6 – Структурна схема додатка

2.2 Підсистема голосового інтерфейсу

Реалізувати самостійно розпізнавання мови не має сенсу, тому що на ринку вже існують системи, відкриті для програмістів. Дані системи використовують швидкі алгоритми і нейронні мережі для аналізу, тому вибираємо з уже існуючих.

Для реалізації функції розпізнавання голосових команд нам необхідно проаналізувати існуючі технології розпізнавання мови і вибрати серед них ті, які будуть задовольняти перерахованим критеріям.

2.2.1 Аналіз технологій розпізнавання мовлення

Є два підходи до розпізнавання мови.

Перший полягає в онлайн розпізнавання голосу. Такі фірми як Google, Яндекс, Apple, Samsung, Amazon і інші, постійно вдосконалюють свої системи голосового управління розумним будинком. Ці системи найчастіше використовують для своєї роботи штучний інтелект: введена фраза передається на центральний сервер, де обробляється і надсилається користувачеві. Таким чином користувач отримує результат. Такий підхід вимагає наявності постійного Інтернет-з'єднання.

Другий підхід заснований на порівнянні введених слів з наявною базою. Список слів формується програмістом при налаштуванні системи розумний будинок і враховує конкретні побажання власника. У такій системі потрібно чітко формулювати свої запити, так як система не здатна їх інтерпретувати. Наприклад, включити світло в залі, вимкнути телевізор, закрити штори і т.д. Мінусом такого підходу буде складність в додаванні функціоналу без знань мов програмування. Система голосового розпізнавання тісно пов'язана з системою голосового виводу, так як розпізнані команди бажано кудись виводити. Це можна робити за допомогою виведення інформації на дисплей, синтезу мови або спеціально записаних семплів. У системі з онлайн розпізнаванням актуально використовувати синтез мови або дисплей з текстовою інформацією, оскільки

заздалегідь не завжди відомо, який результат буде повернений у відповідь на голосовий запит. У системі, що використовує заздалегідь заготовлену базу слів краще використовувати записані семпли. У такій системі при програмуванні можна пов'язати голосовий запит і голосове повідомлення, що виводиться користувачеві. Причому в цьому випадку повідомлення будуть завжди однаковими. Такий підхід дозволяє створити стабільну і швидко працюючу систему, але на розробку цієї системи піде багато часу.

2.2.2 Критерії оцінки

Під час дослідження API для технологій розпізнавання голосу наголос був поставлений насамперед на: наявність російської мови, коректність результатів розпізнавання і ціну продукту.

Під коректністю розуміють збіг результатів розпізнавання з тестовим набором повідомлень. Набір повідомлень включає в себе фрази: «Привіт, будинок», «Включи світло», «Температура». Дані фрази були повторені кілька разів.

Для вибору технології були проведені тести, що включають в себе написання тестової програми і проведення вимірювань точності і часу розпізнавання. Були розглянуті такі реалізації.

2.2.3 Мовні технології SpeechKit

Комплекс мовних технологій від російської компанії Яндекс, який включає розпізнавання і синтез мови, голосовий активацію і виділення смислових об'єктів в усному тексті. Включає в себе Mobile SDK.

SpeechKit Mobile SDK дозволяє вбудувати розпізнавання і синтез мови, а також голосову активацію Яндекса в ваш мобільний додаток для iOS, Android або

Windows Phone.

Якщо кількість голосових звернень до додатка не перевищує 10 000 в добу, використовувати SpeechKit Mobile SDK можна безкоштовно (базова версія). Якщо запитів виявиться більше, є можливість або оплачувати перевищення ліміту, або перейти на тариф бізнес, в якому немає таких обмежень і доступні додаткові функції - наприклад, створення спеціальних тематичних моделей і унікальних голосів. Безкоштовно для освітніх цілей.

Переваги:

- підтримка російської мови.
- безпека: всі повідомлення передаються в зашифрованому вигляді
- працює на основних мобільних платформах

Недоліки:

- можливі обмеження в роботі програмних продуктів через можливе обмеження кількості звернень;
- можливі перебої в роботі через те, що технологія ще в розробці

2.2.4 Google Cloud Speech API

Технологія розпізнавання голосу від компанії Google.

Google Cloud Speech API дозволяє розробникам перетворювати аудіо інформацію в текст, застосовуючи потужні моделі нейронних мереж в простій для використання API [14].

Є підтримка безлічі пристроїв і програмних продуктів. Є можливість роботи на більшості сучасних операційних системах: Android, iOS, Windows 10.

Цінова політика - 0-60 хвилин в місяць безкоштовно, далі 0,006 \$ за 15 секунд мови. Кожен запит округляється до цифри кратної 15. Перші два місяці безкоштовно [15].

Google Cloud Speech API підтримує більше 110 мов (в тому числі російський). Використання нейронних мереж дозволяє швидко (близько 1-2

секунд) перетворювати голосову модель в текст.

Переваги:

- безкоштовний для використання на платформі Android;
- швидке розпізнавання мови і обробка результатів;
- є підтримка російської мови;
- можливість розпізнавання без використання інтернету на платформі Android;

Недоліки:

- при використанні на платформі відмінною від Android вводиться плата за звернення до сервера;
- закритість платформи;

2.2.5 Технологія Alexa Voice Service

Технологія розпізнавання голосу від компанії Amazon.

Alexa Voice Service (AVS) дозволяє інтегрувати голосового помічника Alexa безпосередньо в програмні продукти. Amazon надає доступ до набору ресурсів для швидкого і легкого створення продуктів з підтримкою Alexa, включаючи API, інструменти для розробки апаратного та програмного забезпечення і документацію. З AVS є можливість додати новий інтелектуальний інтерфейс до своїх продуктів і запропонувати клієнтам доступ до зростаючої кількості функцій Alexa і інтеграції в «Розумні будинки» [18].

Переваги:

- швидке розпізнавання і відгук через використання серверів по всьому світу;

Недоліки:

- не має підтримки російської мови;
- немає безкоштовного використання;

2.2.6 SnowBoy

Технологія розпізнавання голосу з відкритим вихідним кодом під Android, iOS, Windows, JavaScript.

Дана технологія дозволяє запускати розпізнавання мови прямо на пристрої без використання сторонніх серверів. SnowBoy використовує мовні моделі. Для початку розпізнавання необхідно записати голос людини, який вимовляє необхідну фразу для розпізнавання, через це відпадає можливість

Переваги:

- запуск розпізнавання без використання серверів, що дозволяє постійно використовувати розпізнавання на пристрої;
- безкоштовна технологія;

Недоліки:

- вимога складання готових мовних моделей;

2.2.7 Тестування систем розпізнавання мови

Оскільки надійність розпізнавання мови є важливою характеристикою, для вибору реалізації API було проведено тестування. Для це була реалізована тестова програма, в яку включаються функції:

- взаємодії з API;
- виміру часу розпізнавання від запуску до кінцевого результату у вигляді тексту;

Для трьох API, «Мовні технології SpeechKit», «Google Cloud Speech API» і «SnowBoy», були проведені в загальній сумі тридцять тестів. Результати тестування були записані в таблицю 1.

Таблиця 1 – Результати тестування

Назва технології	Середній час розпізнавання після 10 запусків	Точність розпізнавання після 10 запусків
Мовні технології SpeechKit	3.9	Невірні результати повністю співпадаючі з початковим текстом
Google Cloud Speech API	1.6	8 з 10 правильно визначено текстів
SnowBoy	1.2	7 з 10 правильно визначено текстів
Технологія AVS не була задіяна в тестуванні, оскільки не має підтримки російської мови, що порушує критерії		

У результаті тестування було отримано рішення, що SnowBoy найбільш підходить для розпізнавання ключового слова, а Google Cloud Speech API для розпізнавання ключових фраз.

Даний вибір зроблено через те, що для технології Snowboy не потрібні віддалені сервера для розпізнавання, але для кожного слова потрібно створення мовної моделі, тому дана технологія добре підходить для розпізнавання кодового слова, наприклад «Дім». Google Cloud Speech API обраний саме для того, щоб розпізнавати фрази, які налаштовують користувачі, без необхідності кожного разу складати моделі. З проведеного тестування і подальшої обробки результатів було зроблено висновок, що мовні технології SpeechKit дуже повільно реагують, та взагалі не точно показують результат. Технологія Alexa Voice Service, взагалі не була включена в тестування, через те що не має підтримки російської мови. Залишились 2 технології, з яких SnowBoy найбільш підходить для розпізнавання ключового слова, а Google Cloud Speech API для розпізнавання ключових фраз.

2.3 Наявність апаратних модулів, їх структури та особливостей програмування, голосового керування функціональним обладнанням “розумного” будинку

Щоб здійснити зв'язок між додатком і «Розумним будинком» необхідно вибрати протокол, за допомогою якого ми будемо передавати дані.

На сьогоднішній день не існує єдиної платформи та інфраструктури для підтримки всіх пристроїв домашньої автоматизації. На даний момент бездротові мережі представлені безліччю технологій, кожна з яких має ті чи інші переваги. Все що використовуються нині в системах домашньої автоматизації протоколи або створювалися для інших цілей (Wi-Fi для високошвидкісної передачі даних), або ж встигли безнадійно застаріти.

Але жодна з мереж не може повністю задовольнити всі запити розробників і споживачів. У кожної з них є присутнім як ряд плюсів, так і мінусів. Тому все ще відсутня повсюдне універсальне рішення. В результаті, спостерігається високий ступінь фрагментації ринку IoT.

При дослідженні упор був поставлений на простоту реалізації, обслуговування, поширеність протоколу і відкритість стандартів. Розберемо найбільш часто застосовуючі протоколи, їх переваги і недоліки.

2.3.1 MQTT

MQTT (Message Queue Telemetry Transport) - спрощений мережевий протокол, який працює поверх TCP / IP. Використовується для обміну повідомленнями між пристроями за принципом видавець-передплатник.

Ідеальний для використання в контролерах і датчиках, де потрібно невеликий

розмір коду і є обмеження по пропускній здатності каналу .

Протокол MQTT працює на прикладному рівні поверх TCP / IP і використовує за замовчуванням тисяча вісімсот вісімдесят-три порт (8883 при підключенні через SSL). Також, можлива робота через Winsocket, що дозволяється адаптувати його на багато платформи.

Обмін повідомленнями в протоколі MQTT здійснюється між клієнтом (client), який може бути видавцем (publisher), передплатником (subscriber) або брокером (broker) повідомлень [23].

Видавець відправляє дані в брокер, вказуючи в повідомленні певну тему, топик (topic). Передплатники можуть отримувати різні дані від безлічі видавців залежно від підписки на відповідні топіки. Клієнт може бути одночасно і передплатником, і видавцем повідомлення.

Переваги:

- простий у використанні. Протокол є програмний блок без зайвої функціональності, який може бути легко вбудований в будь-яку складну систему;
- відкритий стандарт;
- шаблон проектування видавець-передплатник зручний для більшості рішень з датчиками. Дає можливість пристроям виходити на зв'язок і публікувати повідомлення, які не були заздалегідь відомі або визначені;
- легкий в адмініструванні;
- знижено навантаження на канал зв'язку;
- робота в умовах постійної втрати зв'язку або інших проблем на лінії;
- немає обмежень на формат переданих даних.;

Недоліки:

- немає обмежень на формат переданих даних;
- через не обмеження формату даних можливо невідповідності формату даних у різних виробників обладнання;
- слабка безпека: в основний не використовуються методи захисту для

передачі даних;

2.3.2 Протокол XMPP

XMPP (раніше Jabber) був розроблений для системи миттєвого обміну повідомленнями для зв'язку між людьми за допомогою текстових повідомлень. XMPP означає Extensible Messaging and Presence Protocol, або розширюваний протокол обміну повідомленнями та інформацією про присутність. У протоколі XMPP використовується текстовий формат XML в якості вбудованого типу, забезпечуючи природну зв'язок між людьми. Протокол працює по TCP / IP. Концепція дозволяє використовувати даний протокол в IoT.

Переваги:

- децентралізація: Архітектура мережі XMPP схожа з електронною поштою; хто завгодно може запустити свій власний XMPP-сервер і немає будь-якого центрального сервера
- відкритий стандарт;
- безпека: XMPP сервери можуть бути ізольовані від публічних мереж XMPP (наприклад, у внутрішній мережі компанії) і добре захищені

Недоліки:

- надмірність переданої інформації: як правило, більше 70% трафіку XMPP складають повідомлення про присутність, близько 60% яких є зайвими, що створює зайве навантаження на канал зв'язку;
- неефективність передачі бінарних даних: так як XMPP є, по суті, одним довгим XML-документом, неможливо передати нам не модифіковану двійкову інформацію. В результаті цього, для передачі файлів намагаються використовувати додаткові протоколи, наприклад, HTTP

2.3.3 Протокол AMQP

AMQP (Advanced Message Queuing Protocol) – відкритий протокол для передачі повідомлень між компонентами системи. Основна ідея полягає в тому, що окремі підсистеми (або незалежні додатки) можуть обмінюватися довільним чином повідомленнями через AMQP-брокер, який здійснює маршрутизацію, можливо гарантує доставку, розподіл потоків даних, підписку на потрібні типи повідомлень [25].

AMQP заснований на трьох поняттях:

Повідомлення (message) - одиниця переданих даних, основна його частина (зміст) ніяк не тлумачиться сервером, до повідомлення можуть бути приєднані структуровані заголовки.

Точка обміну (exchange) - в неї відправляються повідомлення. Точка обміну розподіляє повідомлення в одну або кілька черг. При цьому в точці обміну повідомлення не зберігаються. Точки обміну бувають трьох типів:

fanout - повідомлення передається в усі причеплені до неї черги;

direct - повідомлення передається в чергу з ім'ям, що збігається з ключем маршрутизації (routing key) (ключ маршрутизації вказується при відправці повідомлення);

topic - щось середнє між fanout і direct, повідомлення передається в черзі, для яких збігається маска на ключ маршрутизації, наприклад, app.notification.sms.
- в чергу будуть доставлені всі повідомлення, відправлені з ключами, що починаються з app.notification.sms .

Черга (queue) - тут зберігаються повідомлення до тих пір, поки не будуть забрані клієнтом. Клієнт завжди забирає повідомлення з однієї або декількох черг.

Переваги:

- відкритий стандарт;
- розроблений для виключення втрат даних між пристроями
- орієнтований на банківські системи, що дає припущення про його

надійності і безпеки в використанні.

Недоліки:

- складність в реалізації і обслуговуванні

2.4 Висновки до розділу 2

Зваживши всі переваги та недоліки протоколів, було прийнято рішення про використання протоколу MQTT, тому що він має відкритий стандарт, та простий у використанні. Тому для розробки ПО обираємо протокол MQTT. За допомогою нього зможемо налаштувати та запрограмувати взаємодію з «Розумним будинком».

3 ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ, ВПРОВАДЖЕННЯ , ТА БЕЗПЕКИ СИСТЕМ ГОЛОСОВОГО КЕРУВАННЯ ФУНКЦІОНАЛЬНИМ ОБЛАДНАННЯМ “РОЗУМНОГО” БУДИНКУ

Основними модулями розробляється є такі:

- підсистема розпізнавання мови;
- підсистема розпізнавання мови;
- підсистема призначеного для користувача інтерфейсу;
- підсистема виконання команд;
- підсистема зберігання даних;

Реалізація виконана на мові Java в середовищі Android Studio. Весь код мобільного додатка знаходиться в одній директорії «MyHome». Її структура виглядає наступним чином:

1) Каталог «Application» - містить вихідний код програми і необхідні для роботи програми файли:

- каталог «MyHome \ app \ src \ main \ java \ com \ alimovnin \ myhome» містить вихідний код мобільного додатка
- AppCompatActivity.java - допоміжне опис засобів налаштувань;
- CommandHandler.java - методи виконання команд;
- Database.java - опису баз даних;
- GoogleRecognize.java - взаємодія з Google Cloud Speech API;
- MqttClient.java - взаємодія с API Paho MQTT;
- MyHomeActivity.java - опис головної сторінки інтерфейсу, що включає в себе функції обраного;
- MyHomeService.java - опис служби для взаємодії з ОС Android, який запускає все додаток
- HotwordRecognize.java - опис взаємодії з API SnowBoy;
- SettingsActivity.java - опису інтерфейсу налаштувань;
- TopicPreference.java, TopicsActivity.java, TopicsAdapter.java;

- опису інтерфейсу налаштувань топіка;
- TTSHandler.java - опису взаємодії з технологією Text-to-speech;
- каталог «MyHome \ app \ src \ main \ res» - містить файли опису інтерфейсу і всього тексту, що виводиться на екран.

2) Каталог «MyHome \ app \ src \ main \ java \ ai \ kitt \ snowboy» містить файли необхідні для роботи SnowBoy.

Файл MyHomeService містить службу, що виконує завдання, для яких не потрібен графічний інтерфейс. Служба виділяє місце в пам'яті для служб розпізнавання, генерації голосу, підключення та взаємодії з MQTT.

3.1 Реалізація голосового інтерфейсу

Для реалізації розпізнавання команд користувача був використані технології SnowBoy і Google Cloud Speech API [пункт 2.2.7]. Технологія SnowBoy працює без підключення до інтернету. Google Cloud Speech API може працювати як з використанням інтернету, так і без нього, за умови встановленого мовного пакета в ОС Android.

3.1.1 Алгоритм роботи голосового інтерфейсу

Для роботи технології SnowBoy були створені мовні моделі слів «MyHome», «Home», «Дім», "Привіт, будинок» [пункт 2.2.6].

Після підключення до «Розумному будинку» відбувається запуск постійного розпізнавання «ключового слова», яке призначено в настройках. Дане розпізнавання працює до припинення з'єднання з MQTT або припинення роботи програми. Якщо ключове слово було визначено, то відбувається запуск розпізнавання ключової фрази. Інформація в текстовому вигляді передається в

підсистему виконання команд.

3.2 Реалізація інтерфейсу до апаратної частини

Для взаємодії с «Розумним будинком» був обраний протокол MQTT [пункт 2.4]. В якості відкритої реалізації протоколу для ОС Android узятий проект *PaHo MQTT*. Даний проект поєднує в собі весь необхідний функціонал для взаємодії з брокером MQTT. Вся інформація, що надходить через API зберігається в базі даних для подальшого аналізу і використання..

3.2.1 Взаємодія з протоколом

Для підключення додатки до MQTT брокеру використовується метод `connect ()` в які передаються параметри введені «Адміністратором»:

- 1) Адреса підключення.
- 2) Ім'я користувача.
- 3) Пароль.
- 4) Параметри часу затримки передачі і обраного початкового топіка.

Після підключення додаток (передплатник) починає зчитувати з брокера всі повідомлення виду назви топіка і значення, після чого передає їх в базу даних для зберігання. Для відключення викликається метод `disconnect ^)`, де відбувається відписка від початкового топіка і призвести до втрати з'єднання з брокером.

3.3 Реалізація бази даних

Для зберігання призначених для користувача даних і топіків MQTT використовується вбудована в Android база даних *SQLite 3*. Дана база даних має функції створення таблиць, створення і зміни існуючих даних і використання пошуку за необхідними параметрами. У базі даних були створені 4 таблиці.

Кожна таблиця має на увазі своє власне призначення:

- 1) Для зберігання даних MQTT (таблиця mqtt).
- 2) Для зберігання кодових фраз (таблиця keyphrase).
- 3) Для зберігання параметрів команд (таблиця parameters).
- 4) Для зберігання параметрів синтезу мовлення (таблиця tts).

Для кожного топика MQTT в базі даних створюється запис з певним id, який зіставляється з даними топилом у всіх таблицях, що дозволяє не створювати повторні записи з даними топилом і відсівати їх за потрібними параметрами.

Для зберігання даних MQTT була створена таблиця mqtt, в ній міститься:

- 1) Назва топика.
- 2) Поточне значення.
- 3) Дата будуть змінені.
- 4) Призначена команда.
- 5) Додаткові параметри: призначене для користувача назва топика, відображення топика.

Для зберігання кодових фраз була створена таблиця keyphrase, в даній таблиці міститься задані користувачами кодові фрази для розпізнавання.

У таблиці parameters містяться числові або символічні значення, на які будуть змінюватися значення топиків MQTT після виконання дій.

Таблиця tts призначена для зберігання:

- 1) Призначену команди для синтезатора мови.
- 2) Додатковий текст при програванні синтезованого тексту.
- 3) Період спрацьовування.
- 4) Додаткові параметри.

3.3.1 Описання роботи бази даних

Для реалізації бази даних був створений клас Database, який дозволяє створювати базу даних, викликом конструктора Database () або очищати вже створену викликом методу clear ().

База даних формується за допомогою методу onCreate, при виклику якого, створюються чотири таблиці - mqtt, keyphrase, parameters, tts.

3.4 Реалізація підсистеми виконання команд

У даній підсистемі відбувається пошук розпізнаної текстової інформації в базі даних, а також виконання всіх необхідних дій зі значенням топіка MQTT. Надалі змінені значення передаються «Розумному будинку». На рисунку 3.1, представлена блок-схема підсистеми виконання команд, також в додатку А наведено код виконання команд.

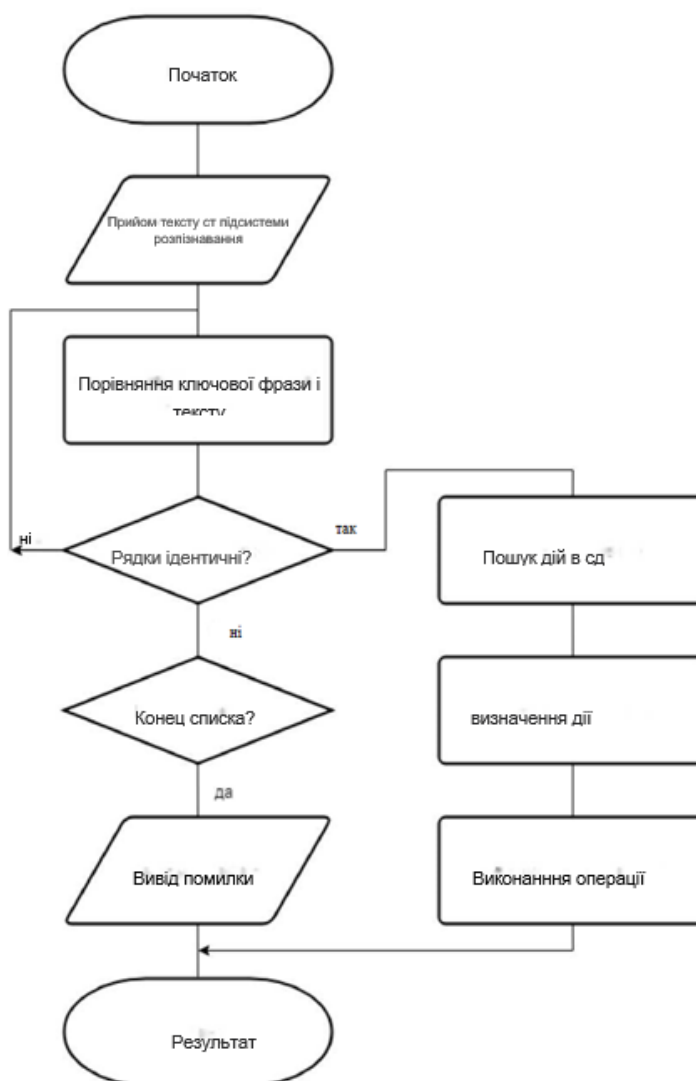


Рисунок 3.1 - Підсистема виконання команд

3.5 Висновки до розділу 3

З проведеного тестування і подальшої обробки результатів було зроблено висновок, що мовні технології SpeechKit дуже повільно реагують, та взагалі не точно показують результат. Технологія Alexa Voice Service, взагалі не була включена в тестування, через те що не має підтримки російської мови.

Залишились 2 технології, з яких SnowBoy найбільш підходить для розпізнавання ключового слова, а Google Cloud Speech API для розпізнавання ключових фраз.

4 ПРИКЛАДИ ВИКОРИСТАННЯ СИСТЕМ ГОЛОСОВОГО КЕРУВАННЯ ФУНКЦІОНАЛЬНИМ ОБЛАДНАННЯМ РОЗУМНОГО БУДИНКУ, ТА ІНТЕРФЕЙС СИСТЕМИ

Для взаємодії з додатком був розроблений для користувача інтерфейс. Функції інтерфейсу розділені на дві категорії:

- 1) Функції «Адміністратора».
- 2) Функції «Клієнта».

Мобільний додаток можна умовно розділити на три основні екрани призначеного для користувача інтерфейсу і одну фонову задачу:

- 1) Екран «Вибране».
- 2) Екрани налаштувань.
- 3) Повідомлення в панелі повідомлень ОС Android.
- 4) Фонова завдання з розпізнавання та генерації голосу.

Мобільний додаток можна використовувати в двох локалізаціях мови: «російський» і «англійська». Дане рішення дозволяє розширити призначену для користувача аудиторію програми.

4.1 Інтерфейс адміністратора

В інтерфейс для користувача «Адміністратор» входить:

- функція налаштування програми;
- функція настройки підключення до MQTT;
- функція настройки топіків;

На рис. 4.1 наведено діаграму сценарію підключення і налаштувань топіків.

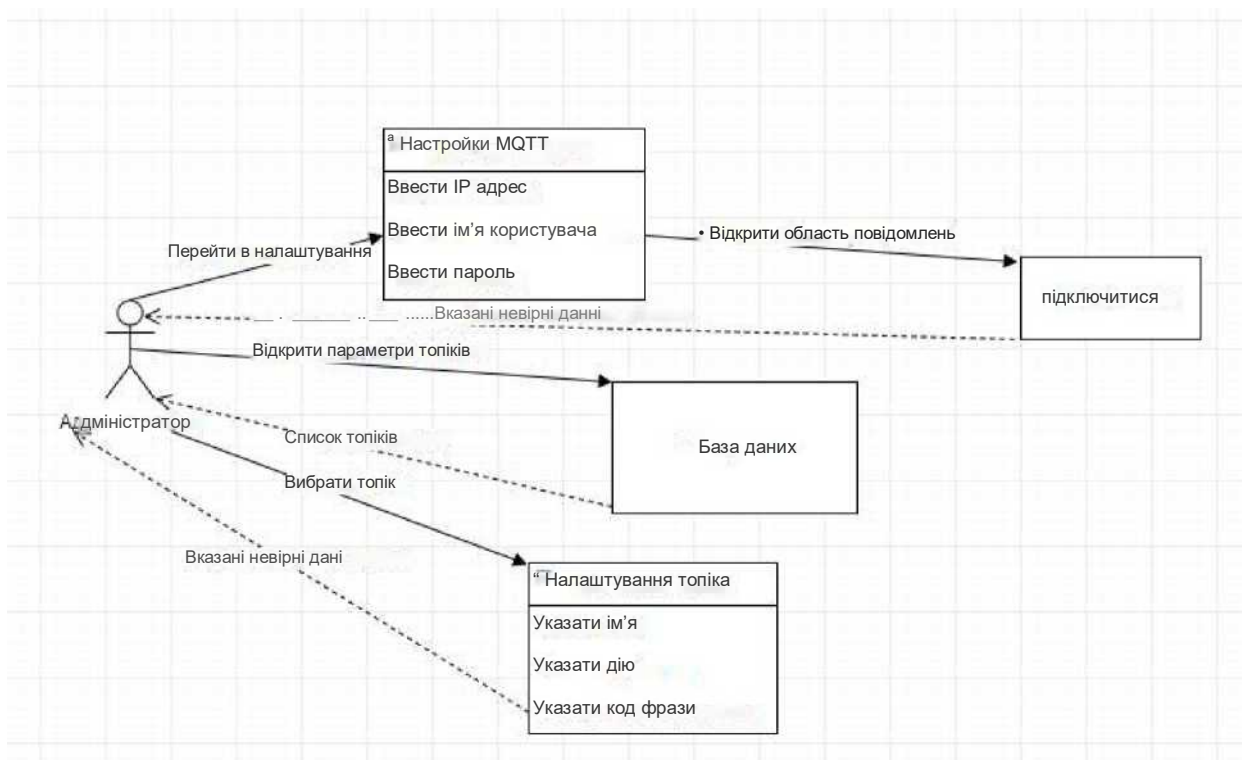


Рисунок 4.1 - Сценарій використання програми користувачем «Адміністратор»

4.1.1 Налаштування додатку

Зовнішній вигляд основного екрану налаштувань надано на рисунку 4.2.

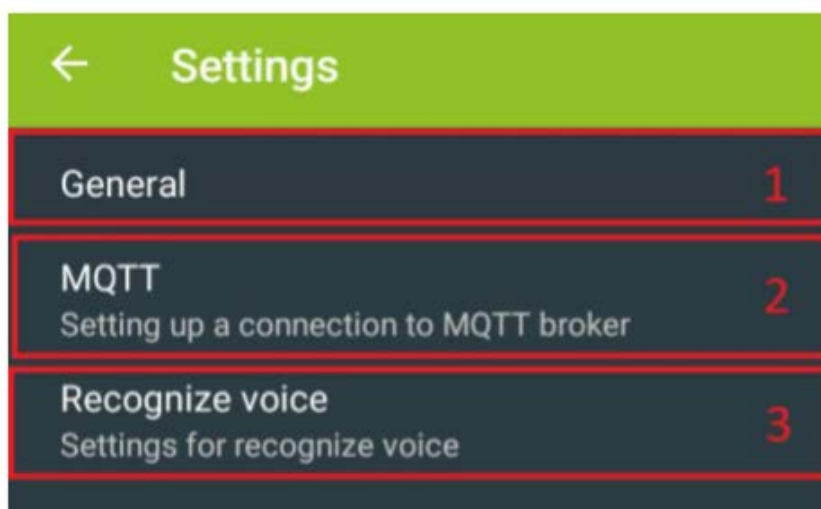


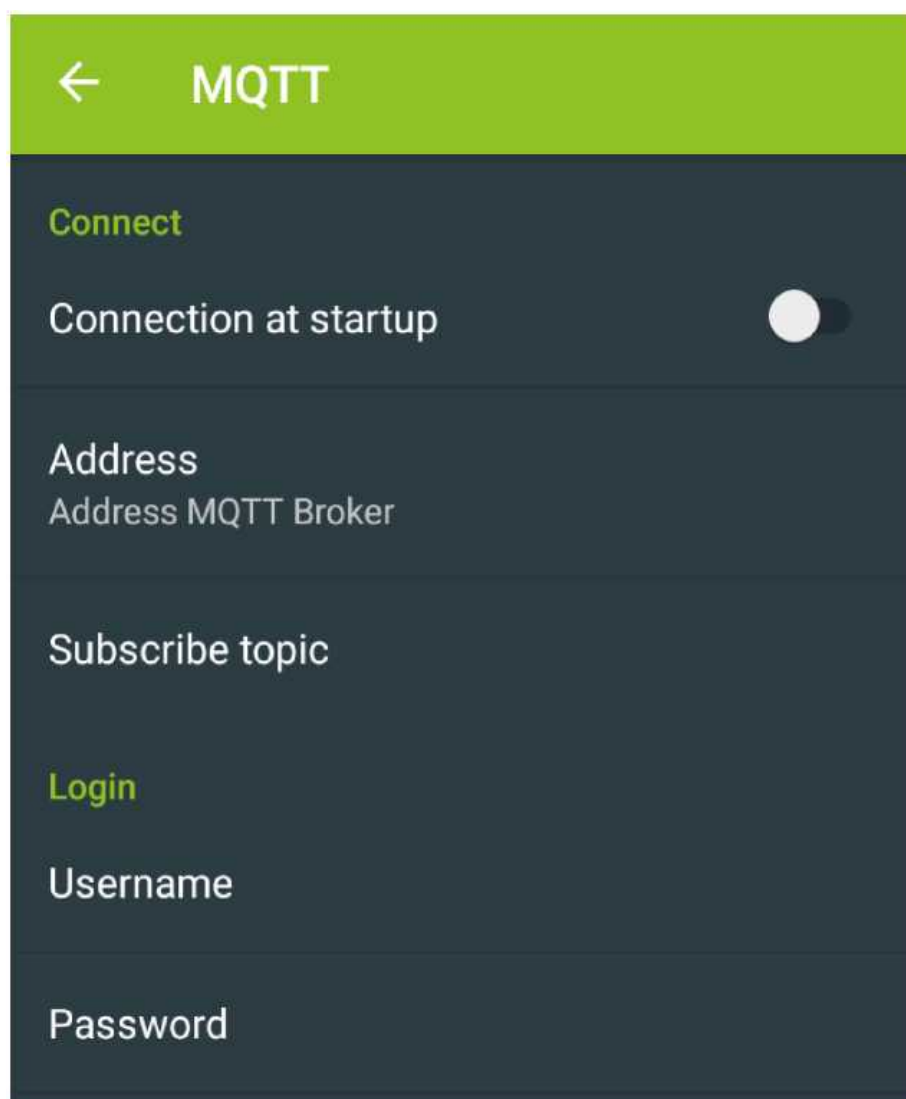
Рисунок 4.2 – Екран налаштувань

Екран розділений на три пункти, цифрами позначені:

- 1) Установки програми.
- 2) Налаштування MQTT.
- 3) Налаштування розпізнавання і генерації голосу.

4.1.2 Налаштування MQTT

Пункт «Налаштування MQTT» містить в собі поля для введення адреси підключення, початкового топіка, імені користувача та пароля. Дані поля необхідно заповнити «Адміністратору» для подальшої коректної роботи програми. Ці настройки показані на рисунку 4.3.



← MQTT

Connect

Connection at startup ☐

Address
Address MQTT Broker

Subscribe topic

Login

Username

Password

Рисунок 4.3 – Екран налаштування MQTT

4.1.3 Налаштування додатку

Пункт «Установки програми» містить в собі три пункти:

- 1) Список всіх топіків.
- 2) Налаштування значень за замовчуванням.
- 3) Скинути базу даних.

Дані налаштування продемонстровані на малюнку 4.4.

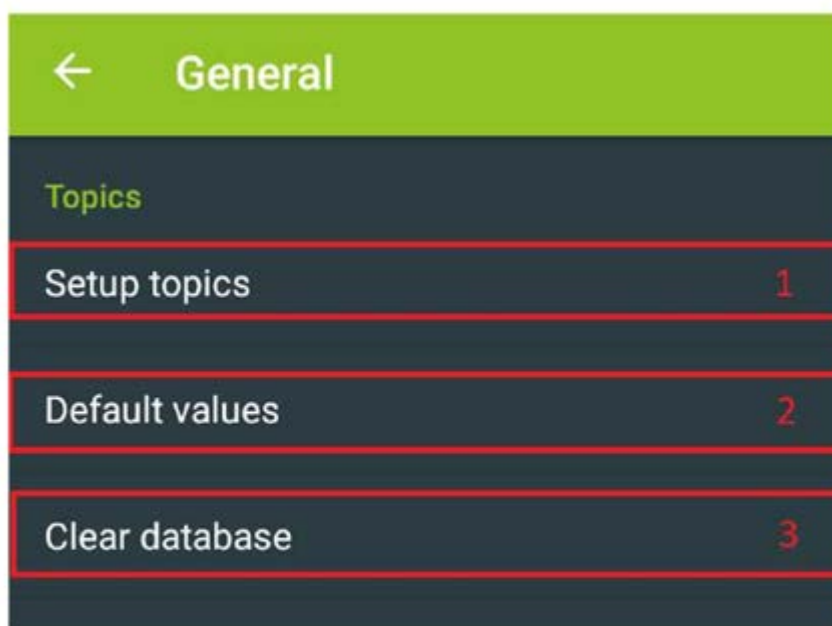


Рисунок 4.4 - Налаштування

4.1.3.1 Налаштування топіків

Пункт «Топіки», зображений на рисунку 4.5, містить екран всіх зберігаються записів в базі даних

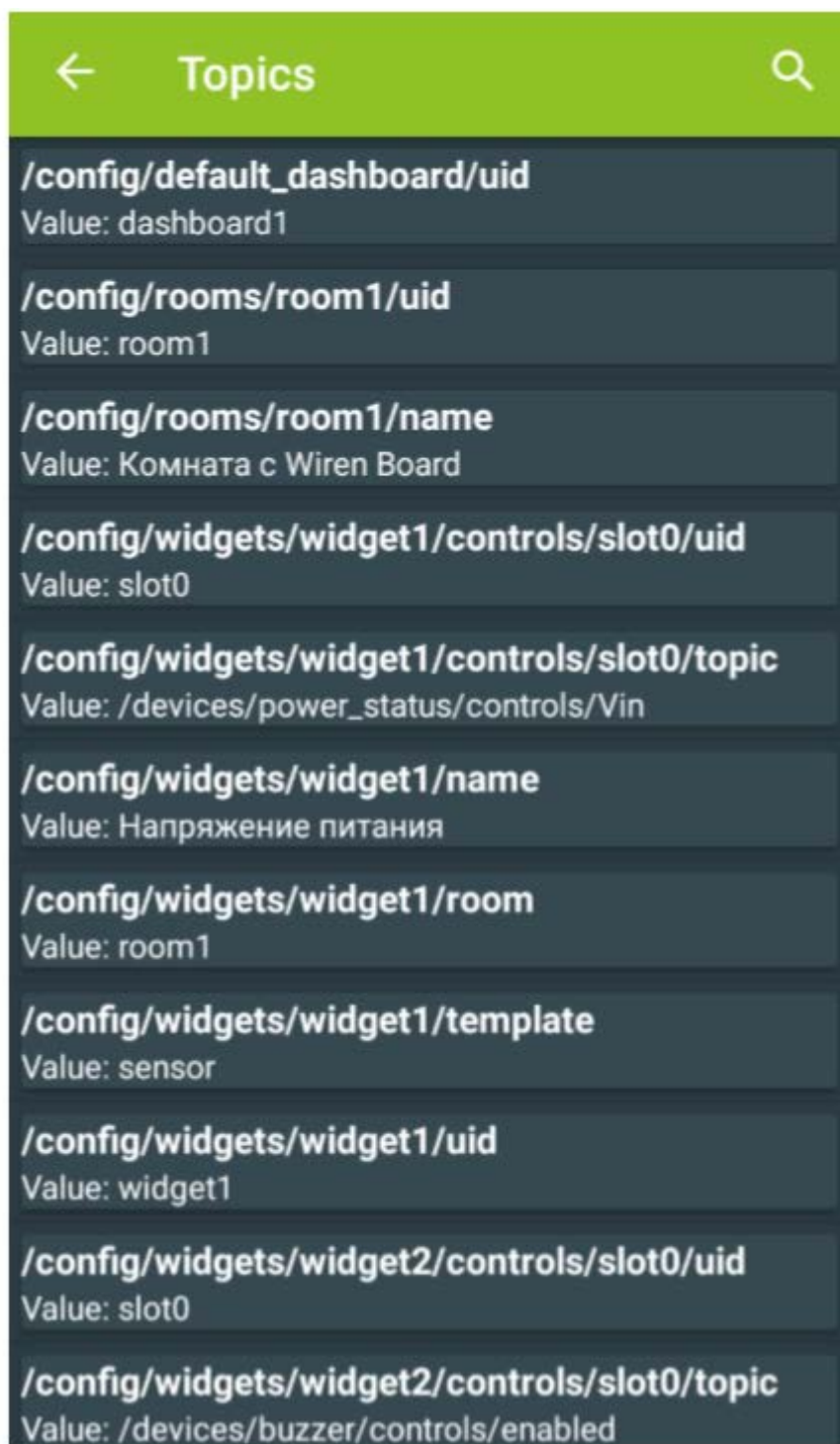


Рисунок 4.5 - Экран топиків

При виборі будь-якого запису відбувається перехід до налаштувань конкретного топика. Екран налаштування топика надано на рисунку 4.6.

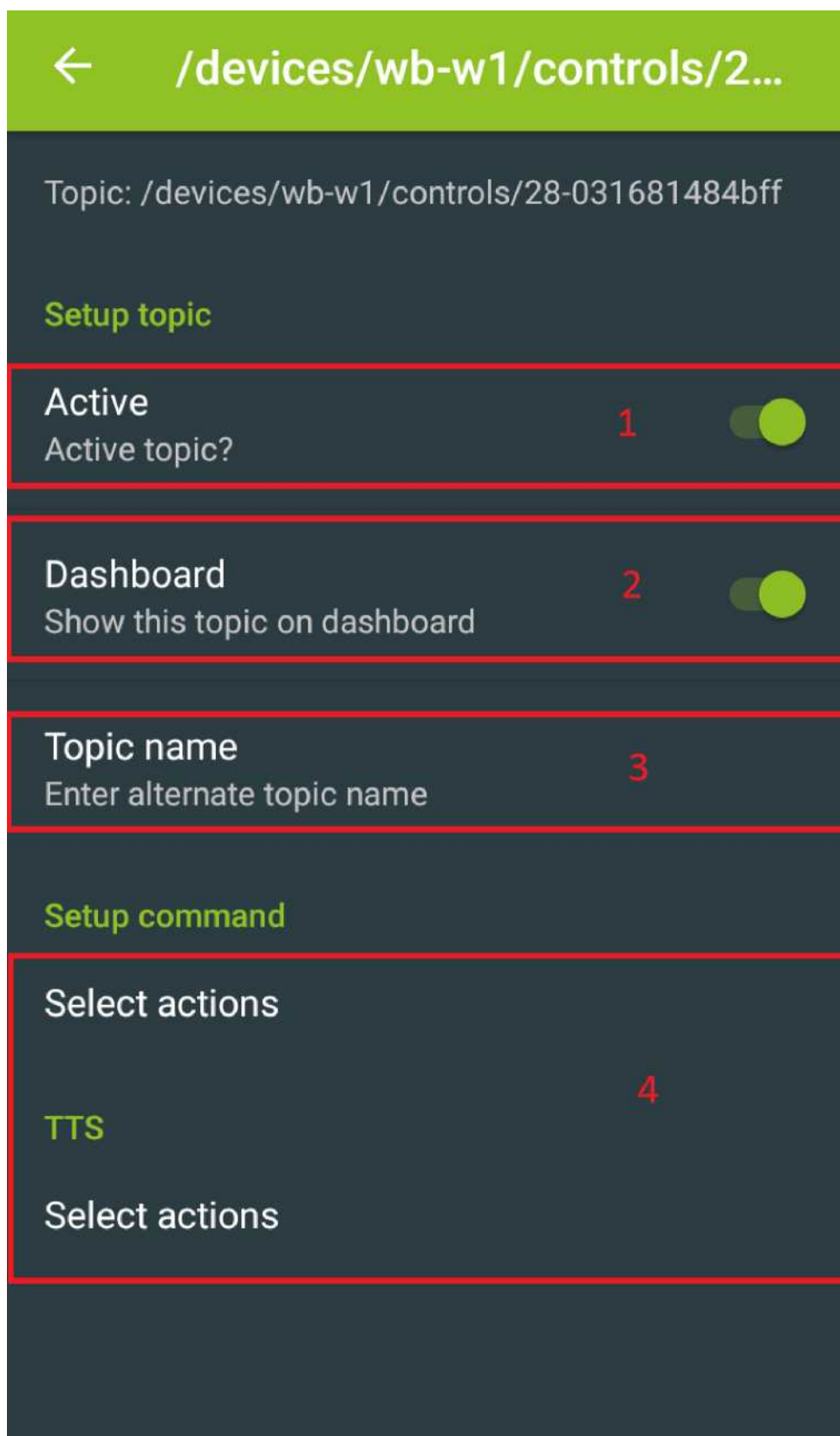


Рисунок 4.6 - Екран налаштування топіка

На цьому екрані користувач з роллю «Адміністратор» може налаштувати:

- 1) Чи показувати цей запис на екрані топіків.
 - 2) Чи показувати цей запис в обраному.
 - 3) Ім'я топіка.
 - 4) Пункти завдання команд для розпізнавання і «дій», які будуть виконані після розпізнавання.
 - 5) Пункти і параметри для налаштувань користувачем «Адміністратор».
 - 6) Пункт «Вибрати дію» містить в собі, що випадає "дій». У додатку заздалегідь включено чотири види «дій»: «Включити-вимкнути», «Включити через час», «Включити і вимкнути через час» і «Збільшити або зменшити значення», після вибору «дії» з'являються додаткові поля для введення команд (-и) і параметрів (-а) в залежності від того, що робить «дія»:
- «включити-вимкнути» - включає в себе два поля введення команд для розпізнавання і два поля для введення значень, які будуть передаватися по протоколу MQTT. Передані значення можуть бути задані користувачем «Адміністратор» в пункті налаштувань «Значення за замовчуванням», змінені для конкретного топіка в поле вводу значень або можуть використовуватися стандартні значення, прописані в програмі:
 - «включити через час» - включає в себе поле введення команди, поле для введення часу і поле для введення значення;
 - включити і вимкнути через час» - включає в себе поле введення команди, поле для введення часу і два поля введення значень включення і виключення
 - «збільшити або зменшити значення» - включає в себе сім полів для введення. Чотири поля призначені для вказівки команд: «Збільшити», «Зменшити», «Встановити максимальне значення», «Встановити мінімальне значення»;
 - Три поля відведено для установки параметрів максимального, мінімального значення, на яку потрібно змінити значення топіка;

7) Пункт «Вибрати дію для TTS» містить в собі, що випадає "дій» для генерації голоси і включає в себе два «дії»:

- «сказати значення один раз» - включає в себе поле введення команди і поле введення тексту для синтезу мови. Цей текст і значення з топіка буде сказано після розпізнавання заданої команди;
- «сказати значення через ...» - включає в себе чотири поля введення. Два поля відведено для команд «Запустити таймер» і «Зупинити таймер», поле введення тексту для синтезу мови, і поле введення часу. Цей текст і значення з топіка буде сказано з певною періодичністю. Запуск і зупинка відбувається після розпізнавання команди;

4.1.5 Налаштування значень за замовчуванням

Для зміни значень за замовчуванням використовується пункт «Значення за замовчуванням» зображений на рисунку 4.7.

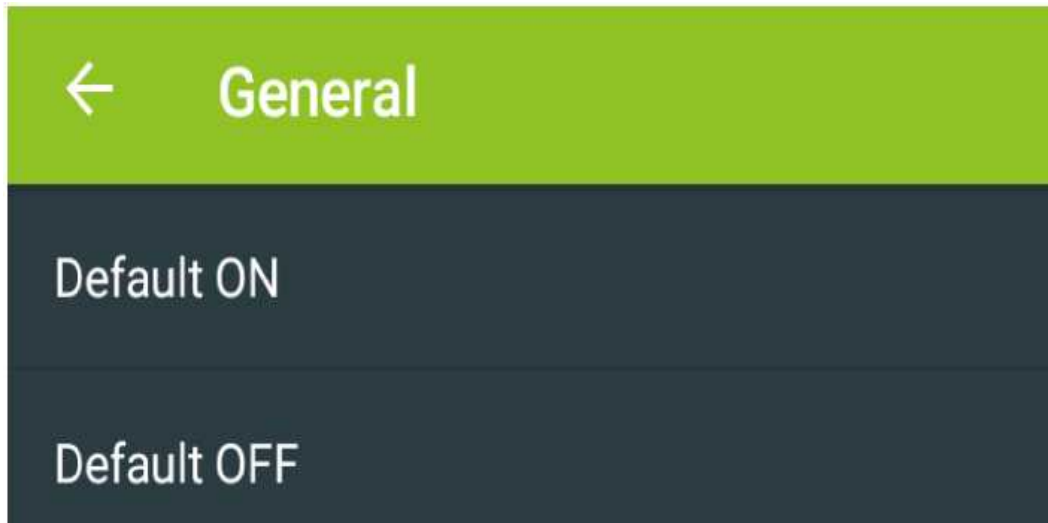


Рисунок 4.7 – Налаштування значень за замовчуванням

4.2 Інтерфейс клієнта

Для зміни значень за замовчуванням використовується пункт «Значення за замовчуванням» зображений на малюнку 4.8.

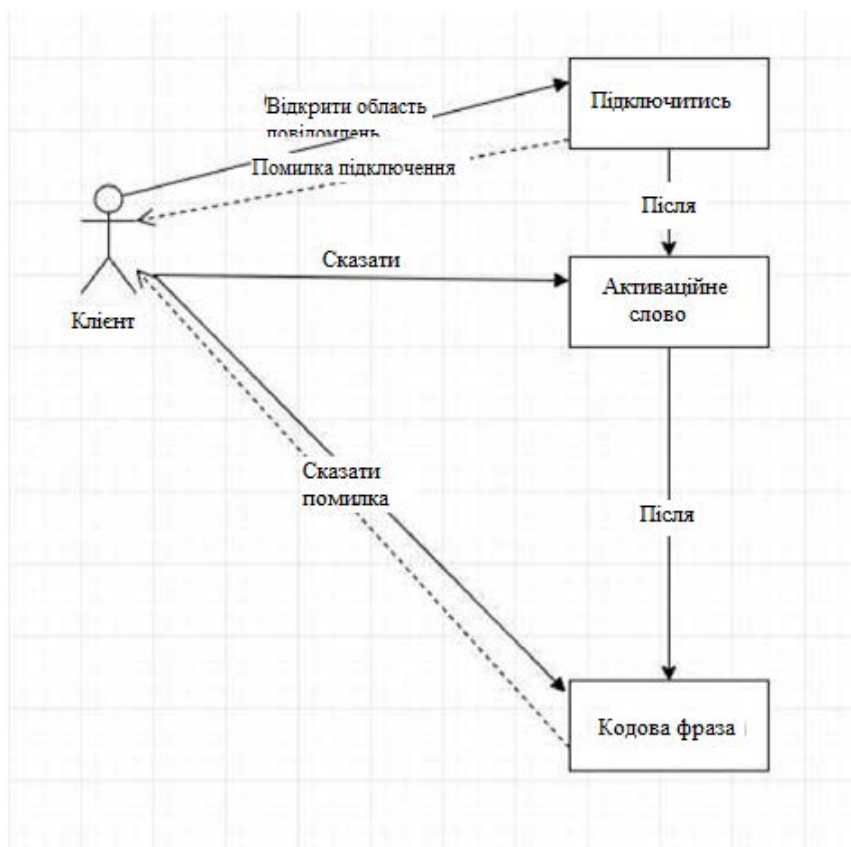


Рисунок 4.8 - Сценарій користувача «Клієнт»

В інтерфейс для користувача «Клієнт» входить:

- функція настройки обраних топіків;
- функція підключення до MQTT, за умови, що користувач з роллю «Адміністратор» налаштував до цього підключення;
- функція подальшого використання: вимова фрази і отримання зворотного зв'язку у вигляді синтезу мови або спливаючого повідомлення.

4.2.1 Обране

Зовнішній вигляд обраного надано на рисунку 4.9. Цифрами позначені:

- 1) Кнопка виклику меню
- 2) Список обраного



Рисунок 4.9 - У головному вікні

Список обраного включає в себе найпопулярніші канали для «Адміністратора» і «Користувача» топиків MQTT. Кожен запис складається з призначеного користувачами назви топика і його поточного значення.

Меню обраного містить два пункти: «Налаштування» і «Про програму». Кнопка «Налаштування» дозволяє перейти в налаштування програми і MQTT. Спливаюче вікно «Про програму» містить в собі інформацію про розробника програми і коротку допомогу в початковому налаштуванні. Дане вікно надано на рисунку 4.10.

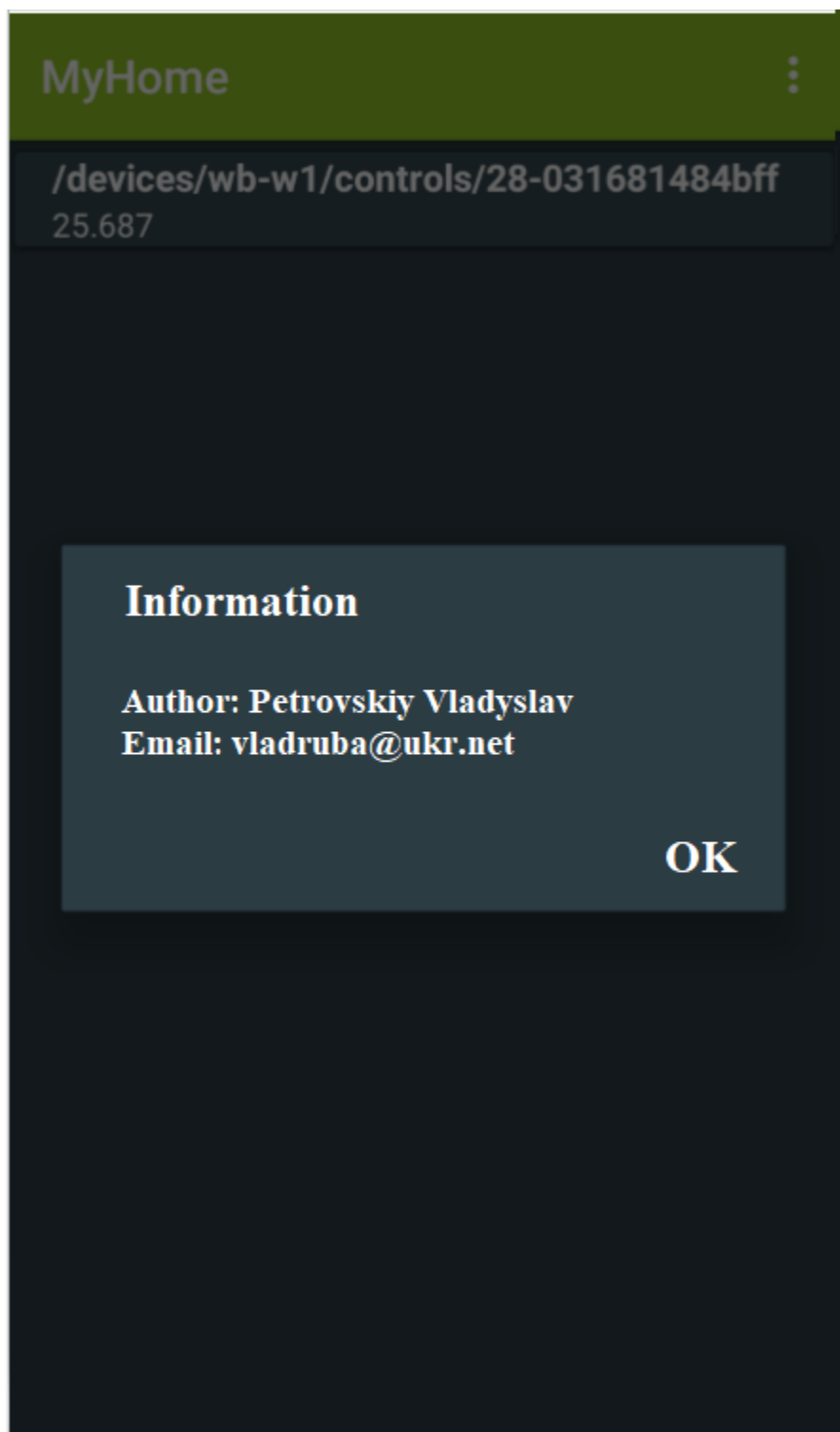


Рисунок 4.10 – Про додаток

4.2.2 Інтерфейс підключення

Інтерфейс для підключення являє собою повідомлення в області повідомлень ОС Android.



Рисунок 4.11 – Повідомлення при відсутності з'єднання з MQTT

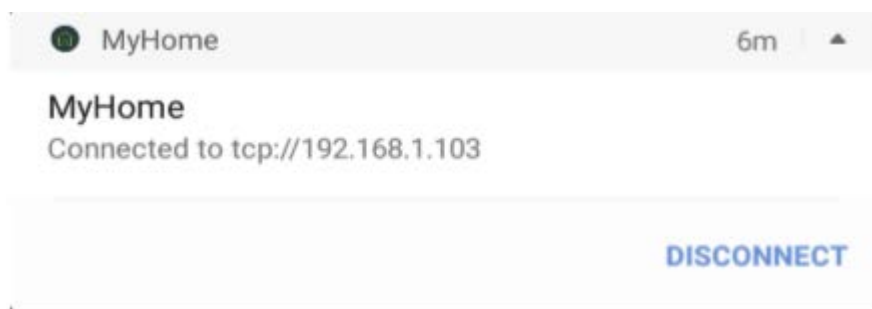


Рисунок 4.12 – Повідомлення при підключенні до MQTT

При натисканні кнопки «Приєднатися» або «Від'єднатися», в залежності від статусу підключення, відбувається зміна повідомлення.

4.3 Тестування додатку

Для тестування програми були проведені тести в умовах існуючої системи «Розумний будинок». Користувачем "Адміністратор» були проведені такі тести.

Налагодження підключення, з введенням коректного і помилкового адреси підключення. В результаті введення помилкового адреси підключення не було скоєно і вивелася помилка підключення. Для коректного адреси з'єднання було успішно встановлено.

Налаштування топіків MQTT з введенням однакових ключових фраз, в результаті програма повідомила про помилку, що неможливо використовувати ключову фразу двічі.

Для користувача «Клієнт» був проведений такі тести.

Налаштування двох топіків, що відповідають за температуру і включення або виключення світла.

Проголошення ключового слова «Дім» і ключової фрази для кожного тесту. Для топика, що відповідає за температуру всередині приміщення, було вибрано дію «Сказати значення один раз» і призначена фраза «Скажи температуру». Для топика, що відповідає за світло, було вибрано дію «Включіть- вимкнути» і призначена фраза «Включи».

4.4 Висновки до розділу 4

В ході тестування програми отримані такі результати, що програма не може правильно визначити кодову фразу при дуже тихої мови. Також були помічені випадкові спрацьовування ключового слова «Дім», але через те, що необхідно ще додати ключову фразу критичного впливу на систему «Розумний будинок» цього не робить. Для того, щоб уникнути випадкового спрацьовування системи було прийнято рішення в подальшому додати функцію підтвердження потенційно небезпечних команд.

ВИСНОВКИ

Завдання щодо дослідження систем голосового керування, які покращують життя людей з обмеженими можливостями, та розроблення мобільного додатка для платформи Android з функцією голосового управління системою «Розумний будинок» виконано і отримано такі результати:

1. Було проведено аналіз існуючих аналогів інтерфейсу голосового управління. Для аналізу були обрані три технології - Google Home, Amazon Echo і Apple Siri. В результаті був зроблений висновок, що технологія Google Home – має високу вартість, але легкий в інтеграції для вже існуючих рішень. Технологія Siri, має безліч переваг, але суттєвий недолік це те, що для роботи з нею потрібні тільки пристрої від компанії Apple. Amazon Echo взагалі не має підтримки російської мови, та також дорогий для інтеграції в системи «Розумний будинок».
2. Проведено збір та аналіз інформації про існуючі технології голосового розпізнавання і взаємодії з «Розумним будинком». Google Cloud Speech API – завдяки нейронних мереж дозволяє швидко перетворювати голосову модель в текст, а також підтримує 110 мов і є безкоштовним для використання на платформі Android. Snowboy, також безкоштовна технологія, але потрібне складання готових мовних моделей. В результаті аналізу і тестування технологій був зроблений висновок, що Google Cloud Speech API найбільш підходить для розпізнавання команд, а SnowBoy для розпізнавання ключового слова.
3. Спроектовано і реалізовано мобільний додаток, що включає в себе функціонал голосового управління системою «Розумний будинок». Він підтримує російську та англійську мови, за допомогою ключової фрази зможе оголосити температуру або включити – вимкнути світло.

4. Додаток було протестовано і показало свою працездатність в реальних умовах системи «Розумний будинок», що дозволяє використовувати реалізоване додаток надалі для користувачів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Мови для «Розумного будинку». URL: <http://www.forbes.ru>. (дата звернення 02.04.2020).
2. Що таке голосові технології // Теплиця соціальних технологій. URL: <http://test.ru> (дата звернення 05.04.2020).
3. Голосовой інтерфейс // Вікіпедія. URL: <http://ru.wikipedia.org> (дата звернення 10.04.2020).
4. Екосистема Amazon Alexa // Geektimes. URL: <http://geektimes.ru> (дата звернення 10.04.2020).
5. Google Home // Вікіпедія. URL: <http://ru.wikipedia.org>
6. Чому «розумні» голосові помічники поки нікому не допомагають // Головні новини. URL: <https://www.gazeta.ru> (дата звернення 14.04.2020).
7. Siri // Національна бібліотека ім Н.Е. Баумана. URL: <https://ru.bmstu.wiki> (дата звернення 15.04.2020).
8. HomeKit // Apple. URL: <https://www.apple.com>
9. Технології розпізнавання мови // ПостНаука. URL: <https://postnauka.ru> (дата звернення 16.04.2020).
10. Надшвидке розпізнавання мови без серверів на реальному прикладі // Habrahabr. URL: <https://habrahabr.ru> (дата звернення 20.04.2020).
11. Вони нас чують: куди розвиваються мовні технології? // Бізнес новини. URL: <http://www.forbes.ru> (дата звернення 20.04.2020).
12. Розпізнавання мови для чайників // Habrahabr. URL: <https://habrahabr.ru> (дата звернення 20.04.2020).
13. Комплекс мовних технологій Яндекс // Яндекс. URL: <https://tech.yandex.ru> (дата звернення 22.04.2020).
14. Cloud Speech-To-Text // Google Cloud
15. Пошук оптимальної системи аудіо розпізнавання мови з закритим вихідним кодом, але мають відкриті API, для можливості інтеграції // Habrahabr. URL: <https://habrahabr.ru>

16. Google оновила Cloud Speech API, додавши підтримку 30 мов // Створено програмістами для програмістів. URL: <https://tproger.ru>
17. Google створив систему розпізнавання мови без підключення до інтернету // Інтернет видання про високі технології. URL: <http://www.cnews.ru>
18. Speak to Alexa on Products with Alexa Voice Service Built-In // Alexa Voice Service. URL: <https://developer.amazon.co>
19. Hotword Detector // Snowboy. URL: <https://snowboy.kitt.ai>
20. Розумний будинок: Розвиток і тенденції // Geektimes. URL: <https://geektimes.ru>
21. Протоколи зв'язку для "розумного будинку" // Аналітичні огляди комп'ютерів. URL: <https://www.ferra.ru>
22. MQTT протокол тепер в пристроях Advantech // Комплексна дистрибуція засобів автоматизації. URL: <http://corson.ru>
23. Що таке MQTT і для чого він потрібен в IoT? // Промислові комп'ютери. URL: <https://ipc2u.ru>
24. Протоколи «Інтернету речей»: основні відомості // Засоби і системи автоматизації. URL: <https://www.rtsoft.ru>
25. RabbitMQ: Введення в AMQP // Habrahabr. URL: <https://habrahabr.ru>
26. MQTT and MQTT-SN messaging protocols // Eclipse Paho. URL: <https://www.eclipse.org>
27. Спеціальні можливості системи "Розумний будинок". Електронний збірник статей переможців V Міжнародної науково-практичної конференції «EUROPEAN SCIENTIFIC CONFERENCE» 30 липня 2017р. в м Пенза. Електронний збірник МК-196 / Алімовнін С.Г., Мільчаков С.А., Сиротинина Н.Ю. // Наука і Просвітництво. - Пенза, 2017. URL: <http://naukaip.ru>
28. Реалізація мовного інтерфейсу системи «Розумний будинок» "Професіонал року 2017" Збірник статей V Міжнародного науково-практичного конкурсу, що відбулася 25 серпня 2017 року в м Пенза. Номер збірки К-56 сторінки 52-56 / Алімовнін З.В., Мільчаков С.А., Сиротинина Н.Ю. // Наука і Просвітництво. URL: <http://naukaip.ru>

29.Протоколи зв'язку для "розумного будинку" // Аналітичні огляди комп'ютерів.

URL: <https://www.ferra.ru>

ДОДАТОК А

Клас обробки команд

```

class CommandHandler { // private ContentValues mContentValues; private Cursor
    mCursor;
    private MQTTClient mMQTTClient;

    CommandHandler(MQTTClient MQTTClient) { mContentValues = new
    ContentValues(); mMQTTClient = MQTTClient;
    }

    void Handler(String result) {//
        //пошук в базі даних
        mCursor = mSQLiteDatabase.query(true, Database.TABLE_NAME
        KEYPHRASE, new String[]{Database.id, Database.phrase 1,
        Database.phrase 2, Database.phrase 3, Database.phrase 4,
        Database.TTSPphrase 1, Database.TTSPphrase_2},
            "UPP      + Database.phr      1 + " LI      '      + res      + "      O      +
            "UPP      + Database.phr      2 + " LI      '      + res      + "      O      +
            "UPP      + Database.phr      3 + " LI      '      + res      + "      O      +
            "UPP      + Database.phr      4 + " LI      '      + res      + "      O      +
        "UPPER(" + Database.TTSPphrase_1 + ") LIKE + result + OR
        " +
            "UPPER(" + Database.TTSPphrase_2 + ") LIKE '%" + result +
        "%'", null, null, null, null, null);
        if (mCursor.getCount() > 0) {//якщо зівпало
            mCursor.moveToFirst();
            if (Objects.equals(result,
            mCursor.getString(mCursor.getColumnIndexOrThrow(Database.phra
            se
            1) ))) {
                Command(mCursor.getString(mCursor.getColumnIndexOrThrow
                (Database.id)), Database.phrase 1);
            } else if (Objects.equals(result,
            mCursor.getString(mCursor.getColumnIndexOrThrow(Database.phra
            se_
            2) ))) {
                Command(mCursor.getString(mCursor.getColumnIndexOrThrow
                (Database.id)), Database.phrase_2);
            } else if (Objects.equals(result,
            mCursor.getString(mCursor.getColumnIndexOrThrow(Database.phra
            se_
            3) ))) {
                Command(mCursor.getString(mCursor.getColumnIndexOrThrow
                (Database.id)), Database.phrase_3);
            } else if (Objects.equals(result,
            mCursor.getString(mCursor.getColumnIndexOrThrow(Database.phra
            se_
            4) ))) {
                Command(mCursor.getString(mCursor.getColumnIndexOrThrow
                (Database.id)), Database.phrase_4);
            } else if (Objects.equals(result,
            mCursor.getString(mCursor.getColumnIndexOrThrow(Database.TTSP
            phrase_1)))) {
                Command(mCursor.getString(mCursor.getColumnIndexOrThrow

```



```

        (Database.id)), Database.TTSPPhrase_1));

    } else if (Objects.equals(result,
mCursor.getString(mCursor.getColumnIndexOrThrow(Database.TTSP
Phrase_2)))) {
        Command(mCursor.getString(mCursor.getColumnIndexOrThrow
(Database.id)), Database.TTSPPhrase_2);
    }
}

//метод обробки
private void Command(String id, String column) { mCursor.close();
mCursor = mSQLiteDatabase.query(true, Database.TABLE_NAME MQTT, new
String[]{Database.id, Database.command, Database.value,
Database.topic},
Database.id + "=?", new String[]{id}, null, null, null, null);
mCursor.moveToFirst();
String value =
mCursor.getString(mCursor.getColumnIndexOrThrow(Database.value));
final String topic =
mCursor.getString(mCursor.getColumnIndexOrThrow(Database.topic));
//

if(mCursor.getString(mCursor.getColumnIndexOrThrow(Database.command
))!=null) {
    //зрівнюємо текст
    switch
    (mCursor.getString(mCursor.getColumnIndexOrThrow(Database.com
mand))) {
        //якщо це дія «Включити-вимкнути» case
        "activate_deactivate":
            mCursor.close();
            MQTT mCursor = mSQLiteDatabase.query(true,
Database.TABLE_NAME PARAMETERS,
new String[]{Database.id, Database.parameters_1,
Database.parameters_2},
Database.id + "=?", new String[]{id}, null, null,
null, null);
            //если действие «Включить»
            if (Objects.equals(column, Database.phrase_1)) {
                value =
                mCursor.getString(mCursor.getColumnIndexOrT
hrow (Database.parameters_1));
            //если действие «Выключить»
            } else if (Objects.equals(column,
Database.phrase_2)) {
                value =
                mCursor.getString(mCursor.getColumnIndexOrT
hrow (Database.parameters_2));
            }
            try {
                // mMQTTClient.publishMessage(topic,
                value);
            } catch (IllegalArgumentException e) {
                e.printStackTrace();
            }
            break;
        }
    }
    //якщо текст було знайдено в базі даних
} else {
    mCursor.close();
}

```

```

//беремо із бд значення MQTT
mCursor = mSQLiteDatabase.query(true, Database.TABLE_NAME
TTS, new String[]{Database.id, Database.TTS,
Database.TTSText}, Database.id + "=?", new String[]{id},
null, null, null, null); mCursor.moveToFirst();
if(mCursor.getString(mCursor.getColumnIndexOrThrow(Database.T
TS) )!=null) {
    switch(mCursor.getString(mCursor.getColumnIndexOrThrow(
Dat abase.TTS))) {

        case "value on command":
            String text =
            mCursor.getString(mCursor.getColumnIndexOrT
hrow (Database.TTSText));
            if(text!=null){
                MyHomeService.mTTSHandler.speak(text
                + " " + value);
            }
            else {
                MyHomeService.mTTSHandler.speak(value);
            }
            break;
        }
    }
}
}
}
}

```

ДОДАТОК Б

SUMMARY

Considering the Smart Home system, there are several types of interfaces for user interaction:

- touch panel with firmware;
- push-button or touch interface in the form of a control panel;
- application that is installed on a user's personal computer or smartphone;

The touch panel is not a very convenient interface, as it is necessary to install it in each room for ease of use, otherwise you need to go to the room where it is installed. The control panel is also not very good in interaction, because it is deprived of the functions of remote control "Smart Home", such as the ability to remotely turn it off or disarming.

Considering the application, you can also identify 2 types of implementation:

- on-screen interface, where the user must click on a certain icon or enter text to perform an action;
- voice interface, which allows you to recognize voice text and perform the necessary actions based on the results;

The voice interface is a software product that allows the user to interact with the Smart Home system using a voice or voice platform, giving commands to run individual scenarios and actions by voice and receiving information in the form of audio messages. Accordingly, the main specific functions of such interfaces are the recognition of voice commands and the synthesis of audio messages.

With the development of technology, the voice interface has become more common, people are increasingly enjoying the benefits of this contactless technology. Voice interfaces are convenient when entering text is difficult or inconvenient. For example, while driving a car, the user can speak his request, dictate the desired address, check traffic jams in the navigator application. Or if the user performs too many tasks and cannot concentrate on one.

A possible solution to implement a voice interface inside the room could be to place microphones and sound sources at different points in the room, so that the user can at any time say the command and hear the answer. However, attempts to implement such systems were faced with the fact that it required a large number of sensitive microphones, so the system became technically complex and expensive. At the same time, modern man almost always carries a mobile phone, which makes the implementation of the voice interface much cheaper, and also adds the possibility of remote control. Therefore, the implementation of a mobile application is a more convenient solution for voice control "Smart Home". The project to develop a mobile application is aimed at the voice control of the "Smart Home".

There are two approaches to language recognition.

The first is online voice recognition. Companies like Google, Yandex, Apple, Samsung, Amazon and others are constantly improving their smart home voice control systems. These systems often use artificial intelligence for their work: the entered phrase is transmitted to a central server, where it is processed and sent to the user. In this way the user gets the result. This approach requires a permanent Internet connection.

The second approach is based on comparing the entered words with the existing database. The list of words is formed by the programmer when setting up the smart home system and takes into account the specific wishes of the owner. In such a system, you need to clearly articulate your queries, as the system is not able to interpret them. For example, turn on the lights in the hall, turn off the TV, close the curtains, etc. The disadvantage of this approach will be the difficulty of adding functionality without knowledge of programming languages. The voice recognition system is closely related to the voice output system, as it is desirable to display the recognized commands somewhere. This can be done by displaying information, speech synthesis or specially recorded samples. In an online recognition system, it is important to use speech synthesis or a display with text information, as it is not always known in advance which result will be returned in response to a voice request. In a system that uses a pre-prepared word base, it is better to use recorded samples. In such a system, when

programming, you can link a voice request and a voice message displayed to the user. And in this case, the messages will always be the same. This approach allows you to create a stable and fast system, but the development of this system will take a long time.

To communicate between the application and the Smart Home, you must select the protocol by which we will transmit data.

To date, there is no single platform and infrastructure to support all home automation devices. Currently, wireless networks are represented by many technologies, each of which has certain advantages. All protocols currently used in home automation systems were either created for other purposes (Wi-Fi for high-speed data transmission), or have become hopelessly outdated.

But none of the networks can fully satisfy all the demands of developers and consumers. Each of them has both a number of pros and cons. Therefore, there is still no universal solution. As a result, there is a high degree of fragmentation of the IoT market.

The study focused on ease of implementation, maintenance, protocol prevalence and open standards. Let's analyze the most commonly used protocols, their advantages and disadvantages.